



A Literature Study On Chef - An Open Source Software Agent

RAJASHEKHAR

M.Tech-Student

Computer Science and Engineering
R V College of Engineering
Bangalore, Karnataka, India

Abstract: Chef is an open source software agent that automates your infrastructure by turning it into code. Our proposal involves using chef for configuring a system for a new employee or employees data to a new system. Presently, after reimage of a system for a new employee or to a new system, configuration of user for some softwares/tools like skype, outlook which needs user data configuration are done manually. CHEF is an automation tool with which we can automate this process by providing user data as parameters for a script. Using CHEF any security patches installation/update of antivirus can be pushed/automated directly to the employee machines. All these can be done with a central CHEF server containing runlist, and a workstation to modify/customise existing scripts to add new features. CHEF also provides a UI interface through which we can manage client machines and their corresponding runlists (components installed using chef). Using this kind of environment reduces manual efforts in configuring new machines for employees. As everything is automated it reduces time for customizing new systems. Using CHEF for automation of user systems configuration saves effort and time thus increasing productivity.

Key word : Open Source Software; Parameters; Script; Customizing; Machines;

I. INTRODUCTION

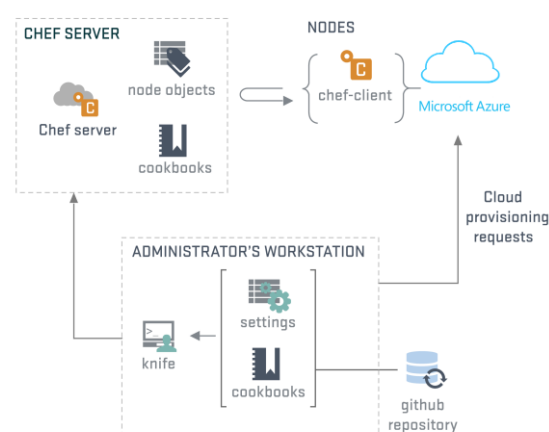
Chef is a company and the name of a configuration management tool written in Ruby and Erlang. It uses a pure-Ruby, domain-specific language (DSL) for writing system configuration "recipes". Chef is used to streamline the task of configuring and maintaining a company's servers, and can integrate with cloud-based platforms such as Internap, Amazon EC2, Google Cloud Platform, OpenStack, SoftLayer, Microsoft Azure and Rackspace to automatically provision and configure new machines. Chef contains solutions for both small and large scale systems, with features and pricing for the respective ranges. Chef was created by Adam Jacob as a tool for his consulting company, whose business model was to build end-to-end server/deployment tools. Jacob showed Chef to Jesse Robbins, who saw its potential after running operations at Amazon. They founded a new company with Barry Steinglass, Nathen Haneysmith, and Joshua Timberman to turn Chef into a product.

The project was originally named "marionette", but the word was too long and cumbersome to type; the "recipe" format that the modules were prepared in led to the project being renamed "Chef". Chef is supported on multiple platforms according to a supported platforms matrix for client and server products. Major platform support for clients includes AIX, RHEL/CentOS, FreeBSD, OS X, Solaris, Microsoft Windows and Ubuntu. Additional client platforms include Arch Linux, Debian and Fedora. Chef Server is supported on

RHEL/CentOS, Oracle Linux, and Ubuntu. Chef is used by Airbnb, Mozilla, Expedia, Facebook, HP Public Cloud, Prezi, Xero, Ancestry.com, Rackspace, Get Satisfaction, IGN, Marshall University, Socrata, University of Minnesota, Wharton School of the University of Pennsylvania, Bonobos, Splunk, Citi, DueDil, Disney, and Cheezburger, among others.

II. ARCHITECTURE

The following diagram depicts the high-level Chef architecture.



Chef has three main architectural components: Chef Server, Chef Client (node), and Chef Workstation. The Chef Server is our management point and there are two options for the Chef Server: a hosted solution or an on-premises solution. We will be using a hosted solution. The Chef Client

(node) is the agent that sits on the servers you are managing. The Chef Workstation is our admin workstation where we create our policies and execute our management commands. We run the **knife** command from the Chef Workstation to manage our infrastructure. There is also the concept of “Cookbooks” and “Recipes”. These are effectively the policies we define and apply to our servers.

III. CONCLUSIONS

Chef as a framework has battle tested generic implementations of how to do many common operations, platform independently where appropriate. Implementing from scratch in regular shell scripts doesn't take advantage of this code reuse.

Chef "converges" on a configuration. A simple custom shell script is less likely to be idempotent and so may fail in an unrecoverable manner under circumstances where another Chef run would work. Of course, a shell script can be written with rigor to achieve the same idempotency, but that further exacerbates the failure to leverage the existing implementations in Chef.

Chef cookbooks can be very shareable, especially when used with a tool like Berkshelf, allowing you to leverage a large library of community contributions. This is more code reuse you lose out on. Between node search, the attribute hierarchy, and data bags, chef provides powerful tools for managing the random differences between multiple deployments of the same application stack. It's very beneficial to have the same logic between various QA, load test, and production environments, while extracting the differences to be a handful of lines that are grouped separately and easily audited.

IV. ACKNOWLEDGEMENTS

The author thanks Dr..S.Sridhar, Professor and Director, Cognitive & Central Computing of R.V.College of Engineering, Bangalore, India for reviewing this literature study and communicating to this Journal for publication. The author also thanks Dr.G.Shobha, Professor and Head, Department of CSE, R.V.College of Engineering, Bangalore, India for the support given.

V. REFERENCES

- [1]. Matt Wrock (2016-09-22). "Chef Client 12.14.89 Released". chef.io. Retrieved 2016-10-10.
- [2]. Jon Morrow (2016-09-26). "Chef Server 12.9.1 Release Announcement". chef.io. Retrieved 2016-10-10.
- [3]. Kumar, Ishant (July 3, 2014). "Understanding Chef and Writing

Cookbooks". TO THE NEW. Retrieved September 2, 2016.

- [4]. Cade Metz (2011-10-26), "The Chef, the Puppet, and the Sexy IT Admin", Wired, retrieved 2015-07-04
- [5]. Alan Sharp-Paul (2013-03-04), Puppet vs. Chef - The Battle Wages On, retrieved 2015-07-04.
- [6]. Lueninghoener, Cory (2011-03-28), "Getting Started with Configuration Management" (PDF), ;login:, Usenix, **36** (2), retrieved 2015-07-04
- [7]. Keiser, John (14 November 2016). "Chef Provisioning: Infrastructure As Code".